Teaching a Computer to Integrate

Joshua Isaacson

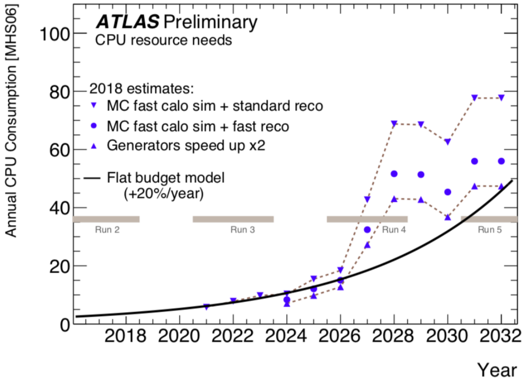
In Collaboration with: C. Gao, S. Höche, C. Krause, H. Schulz

Fermilab AI Jamboree

13 February 2020

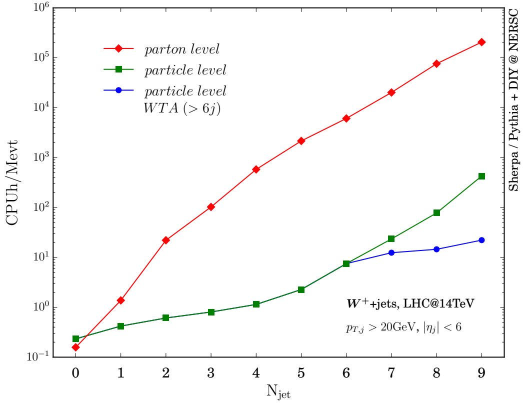
Motivation

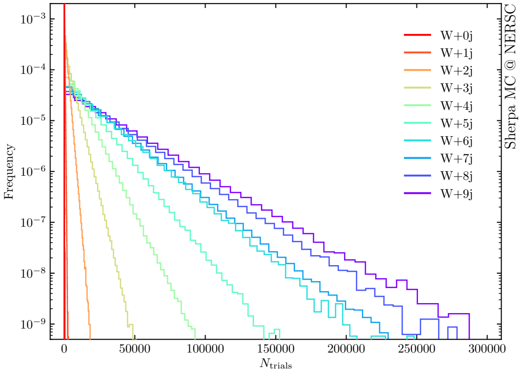
- LHC requires large number of Monte Carlo events
- Due to CPU costs, MC statistics will become significant uncertainty



[ATLAS]

Motivation





[S. Höche, S. Prestel, H. Schulz, 1905.05120]

- Time to generate an event dominated by hard process not shower
- Large computational cost for unweighting at high multiplicity

Importance Sampling

No Importance Sampling

$$\int_0^1 f(x)dx \xrightarrow{MC} \frac{1}{N}\sum_i f(x_i) \quad \text{iid } \mathcal{U}(0,1)$$

Importance Sampling

$$\int_0^1 \frac{f(x)}{q(x)}q(x)dx \xrightarrow{MC} \frac{1}{N}\sum_i \frac{f(x_i)}{q(x_i)} \quad \text{iid } q(x)$$

Importance Sampling

No Importance Sampling

$$\int_0^1 f(x)dx \xrightarrow{MC} \frac{1}{N}\sum_i f(x_i) \quad \text{iid } \mathcal{U}(0,1)$$

Importance Sampling

$$\int_0^1 \frac{f(x)}{q(x)} q(x)dx \xrightarrow{MC} \frac{1}{N}\sum_i \frac{f(x_i)}{q(x_i)} \quad \text{iid } q(x)$$

Goal: Choose a function $q(x)$ such that $\frac{f(x)}{q(x)} \approx 1$.

- Best is $q(x) = f(x)$, requires analytic inverse of CDF
- Acceptable to get close enough by fitting $f(x)$ to some assumed form

Previous Approaches

Generate From Events:

- Pros:
  - Fast evaluation of events
  - Easy to train using existing frameworks
- Cons:
  - Requires large sample of events to train
  - Under-trained $\rightarrow$ Wrong cross-sections

Generate Events:

- Pros:
  - No events required to train
  - Under-trained $\rightarrow$ Still correct cross-section
- Cons:
  - Requires Jacobian of Neural Network in inference

Normalizing Flows

Problem: Numerical Jacobian of Network scales like $\mathcal{O}\left(n^3\right)$
Goal: Develop a network architecture with analytic Jacobian.

Normalizing Flows

Problem: Numerical Jacobian of Network scales like $\mathcal{O}\left(n^3\right)$
Goal: Develop a network architecture with analytic Jacobian.
Requirements:

- Bijective
- Continuous
- Flexible

Normalizing Flows

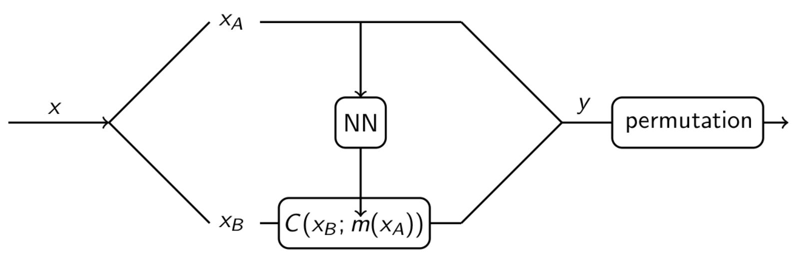
Problem: Numerical Jacobian of Network scales like $\mathcal{O}\left(n^3\right)$
Goal: Develop a network architecture with analytic Jacobian.
Requirements:

- Bijective
- Continuous
- Flexible

Answer: Normalizing Flows!

- First introduced in "Nonlinear Independent Component Estimation" (NICE) [1410.8516]
- More complex transformations using splines in [1808.03856] and [1906.04032]
- Easy to implement using TensorFlow-Probability

Normalizing Flows: Basic Building Block



Forward Transform:

$$y_A = x_A$$

$$y_{B,i} = C(x_{B,i}; m(x_A))$$

Inverse Transform:

$$x_A = y_A$$

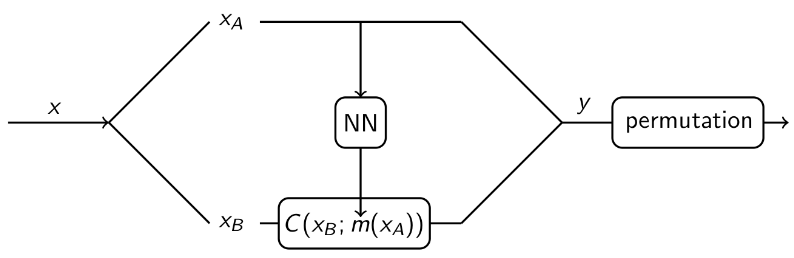
$$x_{B,i} = C^{-1}(y_{B,i}; m(y_A))$$

The C function: numerically cheap, easily invertible, and separable.

Jacobian:

$$\left|\frac{\partial y}{\partial x}\right| = \begin{vmatrix} 1 & \frac{\partial C}{\partial x_A} \\ 0 & \frac{\partial C}{\partial x_B} \end{vmatrix} = \frac{\partial C(x_B; m(x_A))}{\partial x_B}$$

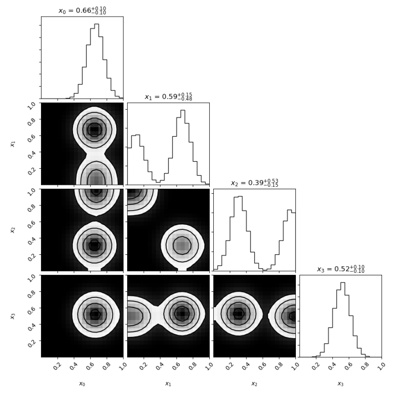
Normalizing Flows: Basic Building Block



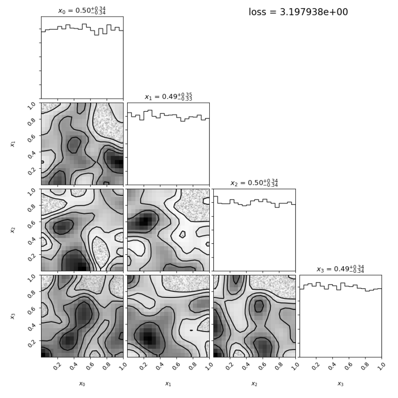
Jacobian is $\mathcal{O}(\mathrm{n})$

Test Functions: 4-d Camel

Target Distribution:



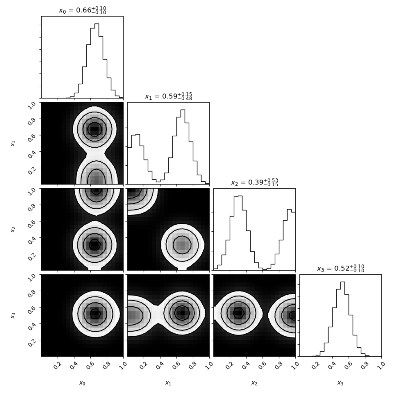
Before Training:



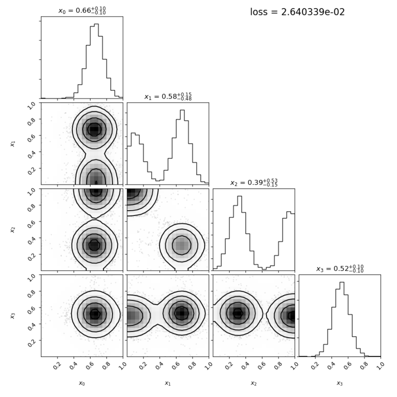
- Final Integral: 0.0063339(41)
- `VEGAS`: 0.0063349(92)

Test Functions: 4-d Camel

Target Distribution:



Final:



- Final Integral: 0.0063339(41)
- VEGAS: 0.0063349(92)

Results

unweighting efficiency $\langle w \rangle / w_{\max}$		LO QCD $n=0$	$n=1$	$n=2$	$n=3$	$n=4$	NLO QCD (RS) $n=0$	$n=1$
$W^+ + n$ jets	Sherpa	$2.8\cdot 10^{-1}$	$3.8\cdot 10^{-2}$	$7.5\cdot 10^{-3}$	$1.5\cdot 10^{-3}$	$8.3\cdot 10^{-4}$	$9.5\cdot 10^{-2}$	$4.5\cdot 10^{-3}$
	NN+NF	$6.1\cdot 10^{-1}$	$1.2\cdot 10^{-1}$	$1.0\cdot 10^{-3}$	$1.8\cdot 10^{-3}$	$8.9\cdot 10^{-4}$	$1.6\cdot 10^{-1}$	$4.1\cdot 10^{-3}$
	Gain	2.2	3.3	1.4	1.2	1.1	1.6	0.91
$W^- + n$ jets	Sherpa	$2.9\cdot 10^{-1}$	$4.0\cdot 10^{-2}$	$7.7\cdot 10^{-3}$	$2.0\cdot 10^{-3}$	$9.7\cdot 10^{-4}$	$1.0\cdot 10^{-1}$	$4.5\cdot 10^{-3}$
	NN+NF	$7.0\cdot 10^{-1}$	$1.5\cdot 10^{-1}$	$1.1\cdot 10^{-2}$	$2.2\cdot 10^{-3}$	$7.9\cdot 10^{-4}$	$1.5\cdot 10^{-1}$	$4.2\cdot 10^{-3}$
	Gain	2.4	3.3	1.4	1.1	0.82	1.5	0.91
$Z + n$ jets	Sherpa	$3.1\cdot 10^{-1}$	$3.6\cdot 10^{-2}$	$1.5\cdot 10^{-2}$	$4.7\cdot 10^{-3}$		$1.2\cdot 10^{-1}$	$5.3\cdot 10^{-3}$
	NN+NF	$3.8\cdot 10^{-1}$	$1.0\cdot 10^{-1}$	$1.4\cdot 10^{-2}$	$2.4\cdot 10^{-3}$		$1.8\cdot 10^{-3}$	$5.7\cdot 10^{-3}$
	Gain	1.2	2.9	0.91	0.51		1.5	1.1

Conclusions

Traditional Integration

- Numerical integration and the need for Monte Carlo
- Current approaches not sufficient for LHC

Conclusions

Traditional Integration

- Numerical integration and the need for Monte Carlo
- Current approaches not sufficient for LHC

Normalizing Flows

- Avoid computational difficulty of Jacobian
- Using splines to approximate CDF

Conclusions

Traditional Integration

- Numerical integration and the need for Monte Carlo
- Current approaches not sufficient for LHC

Normalizing Flows

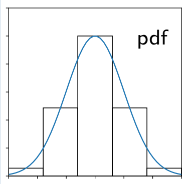
- Avoid computational difficulty of Jacobian
- Using splines to approximate CDF

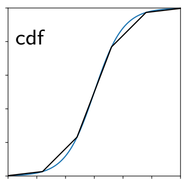
Results

- Better than Sherpa up to $3j$ in all but Z channel
- Room still for further optimization
- Limited by computational resources to train, and not by algorithm

Normalizing Flows: Piecewise CDF

Piecewise Linear CDF: [Müller et al. 1808.03856]

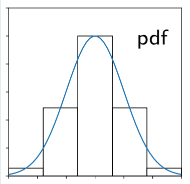


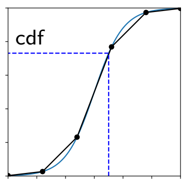


The NN predicts the pdf bin heights Q_i.

Normalizing Flows: Piecewise CDF

Piecewise Linear CDF: [Müller et al. 1808.03856]





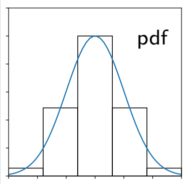
$$C = \sum_{k=1}^{b-1} Q_k + \alpha Q_b$$

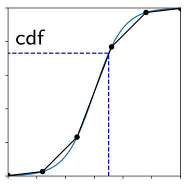
$$\alpha = \frac{x - (b-1)w}{w}$$

$$\left|\frac{\partial C}{\partial x_B}\right| = \prod_i \frac{Q_{b_i}}{w}$$

The NN predicts the pdf bin heights Q_i.

Piecewise Linear CDF: [Müller et al. 1808.03856]





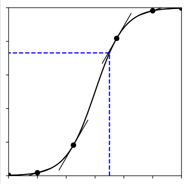
$$C = \sum_{k=1}^{b-1} Q_k + \alpha Q_b$$

$$\alpha = \frac{x - (b-1)w}{w}$$

$$\left|\frac{\partial C}{\partial x_B}\right| = \prod_i \frac{Q_{b_i}}{w}$$

The NN predicts the pdf bin heights Q_i.

Rational Quadratic CDF: [Durkan et. al. 1906.04032]



$$C = y^{(k)} + \frac{(y^{(k+1)} - y^{(k)})[s^{(k)}\alpha^2 + d^{(k)}\alpha(1-\alpha)]}{s^{(k)} + [d^{(k+1)} + d^{(k)} - 2s^{(k)}]\alpha(1-\alpha)}$$

$$\alpha = \frac{x - x^{(k)}}{w^{(k)}} \qquad s^{(k)} = \frac{y^{(k+1)} - y^{(k)}}{w^{(k)}}$$

Predict widths ($w^{(k)}$), heights ($y^{(k)}$), and derivatives ($d^{(k)}$) of the knots of spline.